

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 1 089 495 A2

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:  
04.04.2001 Bulletin 2001/14

(51) Int. Cl.<sup>7</sup>: H04L 12/28, H04L 29/06

(21) Application number: 00308129.6

(22) Date of filing: 18.09.2000

(84) Designated Contracting States:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE

Designated Extension States:

AL LT LV MK RO SI

(30) Priority: 01.10.1999 US 157289 P  
02.08.2000 US 631251

(71) Applicant:

Nortel Networks Limited  
Montreal, Quebec H2Y 3Y4 (CA)

(72) Inventors:

• Wurch, Donald L.  
Rockwall, TX 75078 (US)

- Le, Liem Q.  
Richardson, TX 75081 (US)
- Becker, Carey B.  
Plano, TX 75075 (US)
- Qaddoura, Emad A.  
Plano, TX 75093 (US)
- Coffin, Russ C.  
Plano, TX 75093 (US)

(74) Representative:

Sommerville, John Henry et al  
Sommerville & Rushton,  
45 Grosvenor Road  
St. Albans, Hertfordshire AL1 3AW (GB)

### (54) Method and system for switching between two network access technologies without interrupting active network applications

(57) A system and method is provided for seamlessly switching between different network access technologies without interrupting active network applications or sessions. A Network Access Arbitrator (NAA), which contains a virtual network adapter driver, resides between a Data Link Layer and a Network Layer of the standard OSI-7 Layer Protocol Stack for controlling necessary switching between different network access

technologies. Since all network applications are controlled by layers residing on or above the Network Layer, all applications using network services provided by the Network Layer will continue their active network sessions or applications without disruption, as the NAA switches between different network access technologies.

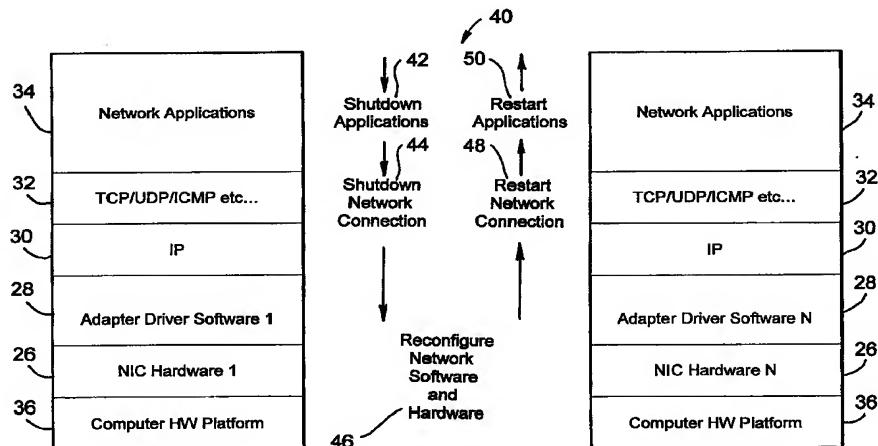


Fig.2

## Description

### Background of the Invention

**[0001]** The present invention relates generally to communication network access technologies, and more particularly, to a system and method for providing transparent and automatic switching between different network access technologies without interrupting active network applications or sessions.

**[0002]** The advent of computer networks has brought a revolutionary change to the world about how people work with computers in their daily activities. Networked computers allow users to share various computer resources and provide significant conveniences to users. Various network access technologies (NATs) are co-existing today that provide users with different network design alternatives. For example, Token Ring, Ethernet, and Wireless Local Area Network are all well known network access technologies that are widely used. Therefore, it is very likely that multiple networks using different network access technologies are located side by side in a larger network that services, for example, a large company.

**[0003]** This co-existence of different network access technologies brings problems and undesired delays when a user switches from one part of a network to another if each part uses different access technologies. For example, it is possible that a user=s laptop is equipped with one Ethernet card and one wireless LAN PCMCIA card for providing two alternate network accesses to a corporate network. For example, consider a user in the middle of an active network session, such as downloading a lengthy file via Ethernet access in the user=s office, and the user must go to a meeting with his colleagues in another building and must bring the file with him. If he has to wait for the file to be completely downloaded, he may be late for the meeting. Alternatively, the user can terminate the downloading session and download the same file all over again at the meeting where he reestablishes a network connection (either through another Ethernet connection in the meeting room or through the wireless LAN PCMCIA connection on his laptop). Both choices are not desirable because either the user is delayed or he has to waste whatever has been downloaded before he leaves his office.

**[0004]** When a user has to disconnect from a network while using a particular NAT and reestablish another network connection through a different NAT, certain processes must happen. In any network that is in conformance with the standard Open Systems Interconnection (OSI)-7 Layer model, all activities in different layers must be terminated.

**[0005]** Referring now to Fig. 1, an overall schematic for the standard OSI-7 Layer Protocol Stack 10 is shown. The concept of layering is generally known in the art and the OSI standard is the only internationally accepted framework of standards for communication

between different system made by different vendors. The OSI-7 Layer Protocol Stack 10 typically has seven different layers: a Physical Layer (L1) 12, a Data Link Layer (L2) 14, a Network Layer (L3) 16, a Transport Layer (L4) 18, a Session Layer (L5) 20, a Presentation Layer (L6) 22 and an Application Layer (L7) 24. As shown in Fig. 1, L1 deals with the physical means of transmitting data over communication lines, and in a network environment, usually refers to various Network Interface Cards (NICs) 26 designed for different NATs. L2 is concerned with procedures and protocols for operating the communication lines, and in this example, is the corresponding Adapter Driver Software 28 for various NICs. In order to identify each NIC, usually a Data Link Layer address or an L2 address is assigned to the NIC. L3 provides information 30 about how data packet routing and relaying can be accomplished. This information may include network or Internet Protocol addresses for communication nodes such as a file server or other computers. L4 defines the rules for information exchange, e.g., information about various network protocols 32 such as TCP/IP protocols, UDP, or ICMP. L5, L6 and L7 are dedicated more to network applications 34. All these layers are working together on a computer hardware platform 36 such as a host computer server.

**[0006]** Now referring to Fig. 2, a flow diagram 40 is shown for terminating a first network access with a first NAT and switching to a second network access with a second NAT, all while active network applications are in progress. When terminating the first network access, the active network applications are interrupted. From the perspective of layering, the active network applications relating to L5, L6 and L7 are first shut down in step 42. Then the corresponding network connections (relating to L4 and L3) are destroyed in step 44. Eventually network software and hardware in L2, L1 and the computer platform are reconfigured in step 46. Using the new NAT, network connections must be initiated in step 48, and the network applications must be restarted again in step 50. In summary, the conventional techniques for switching from the first NAT to the second NAT tears down all processes from L7 downward to L1, and then reestablishes the applications back from L1 upward to L7. This lengthy process incurs extra delays and expenses for network computing and greatly reduces the efficiency of network applications.

**[0007]** What is needed is a method and system to switch between different network access technologies without interrupting active network applications or sessions.

### Summary of the Invention

**[0008]** A system and method is provided for seamlessly switching between different network access technologies without interrupting active network applications or sessions.

**[0009]** Using as an example the standard OSI-7 Layer Protocol Stack to implement network communications, one embodiment of the present invention provides a Network Access Arbitrator (NAA). The NAA is a virtual network device driver situated between the Data Link Layer (L2) and the Network Layer (L3) of the OSI-7 Layer Protocol Stack for controlling necessary switching between different network access technologies. Since all computer network applications are controlled by layers residing on or above L3, all applications using network services provided by L3 (connection or connectionless) will continue their active network sessions without disruption, as the NAA switches between network access technologies.

**[0010]** In addition to providing switching between different network access technologies, the NAA also works in conjunction with Mobile Internet Protocol functions such as IP-in-IP encapsulation/de-capsulation, proxy ARP, gratuitous ARP, etc.

#### Brief Description of the Drawings

**[0011]**

Fig. 1 illustrates an overview of the standard OSI-7 Layer Protocol Stack.

Fig. 2 illustrates a process flow for switching between two different network access technologies.

Fig. 3 is a graphical representation of how a Network Access Arbitrator interacts with different Layers of the OSI-7 Layer Protocol Stack in accordance with one embodiment of the present invention.

#### Description of the Preferred Embodiment

**[0012]** Referring now to Fig. 3, a Network Access Arbitrator (NAA) 60 is shown in the environment of an OSI-7 Layer Protocol Stack 10 according to one embodiment of the present invention. The NAA 60 is a virtual adapter driver located between L2 and L3 for providing seamless network hand-offs between two different network access technologies (NATs). With the implementation of the NAA 60, various active network applications are uninterrupted as the NAA 60 stops exchanging information through an existing NAT and moves over to use a new NAT.

**[0013]** On a computer hardware platform 36 such as a host computer server on L2, there are multiple NATs available, e.g., multiple network interface cards (NICs) 62 along with their corresponding adaptive driver software (NIC 0 to NIC N). The NAA 60 is inserted in between L2 and L3. The NAA 60 insures that L3 detects only a virtual Anchor Adapter driver (Anchor) even though there are multiple NICs 62 and adapter drivers installed on the computer platform 36. Therefore, all the processes on and above L3 are not aware of different NICs 62 and adapter drivers 64.

**[0014]** Out of all the available adapters or NICs 62 on the computer platform, one particular NIC can be initially set as a primary adapter. Its driver thus is the primary adapter driver. All the other adapters and their corresponding drivers are considered non-primary or secondary. Initially, the primary adapter driver is the Anchor.

**[0015]** When executing a network application, the primary adapter is usually the one for providing the network access. At any moment, only one of the adapters or NICs is active. However, due to the availability of multiple NATs, the active network adapter may or may not be the primary adapter. The active adapter receives and transmits all Internet Protocol (IP) data packets including those in unicast, multicast, and broadcast format. However, inactive adapters will receive only multicast and broadcast packets. Moreover, the NAA 60 monitors all the adapters 62, and receives and transmits data packets only through the active adapter. Since the NAA 60 is situated between L2 and L3, all network applications or communications using L3 network protocols deal exclusively with the NAA 60 without directly involving any L2 network components. In other words, without letting L3 know which adapter driver in L2 and its associated active adapter in L1 is actually used, the NAA 60 supplies/retrieves data packets to/from the active adapter, whether it is the primary adapter or any other one connected to the same host computer hardware platform. Therefore, an active network application that works with L3 network protocols observes a constant data stream coming from the NAA 60 and sends back to the NAA 60 another data stream for outgoing information without noticing a transition between two NATs.

**[0016]** The NAA 60 treats outgoing data packets and incoming data packets differently. For an outgoing data packet, if the active adapter is the primary adapter, the data packet is sent unmodified from the NAA 60 to the primary adapter except when there is a special need for encapsulation. If the active adapter is an adapter other than the primary adapter, a hardware frame of the data packet is modified by the NAA so that a source hardware address in the frame is set to the L2 address of the active adapter before data packet is sent to that active adapter.

**[0017]** For an incoming packet, if the receiving adapter is the primary adapter, the data packet is passed up@ unmodified to the NAA 60, except when there is a special need for decapsulation. If the receiving adapter is not the primary adapter, a hardware frame of the data packet is modified so the destination hardware address is set to the L2 address of the primary adapter before the data packet is passed through the NAA 60. This ensures that L3 sees no change in the Anchor (that it detects at all time).

**[0018]** In addition, Address Resolution Protocol (ARP) must be blocked or handled appropriately so that an ARP module of the protocol stack is not confused about a single IP address in L3 with multiple L2

addresses. For instance, in response to an ARP request message sent by a router, a message can be broadcasted to publish the L2 address of the active adapter.

**[0019]** Furthermore, it is important for the NAA 60 to determine which network adapter or NIC is active at any moment. Some NICs and their associated adapter drivers are capable of indicating a connection and disconnection status. Typically, the time required to detect a disconnection detection is around one second and around six seconds to detect connection. These time thresholds are good indicators of the activity status of the NICs. The NAA 60 is thus capable of making use of these hardware status indications to obtain information about which adapter is active.

**[0020]** Also, according to one embodiment of the present invention, the NAA 60 is equipped with a timer that times out on a one-second basis. This timed event is used to detect the existence of incoming data packets. If the NAA 60 detects a data packet for the primary adapter, the primary adapter is deemed the active adapter. If the NAA 60 detects that there is no data packet going through the primary adapter in a period of two seconds, but there is at least one data packet received on a non-primary or a secondary adapter, the secondary adapter is used as the active adapter. An active adapter is viewed by the NAA 60 as active until another active adapter replaces it.

**[0021]** With the implementation of the NAA 60, a user can freely switch from one NAT to another without worrying about disrupting any active network applications. For example, as mentioned above, if a user=s laptop is equipped with one Ethernet card and one wireless LAN PCMCIA card, thereby providing for two alternate network access to a corporate network, network application will not be interrupted when the network access is switched from the Ethernet card to the PCMCIA card. The NAA 60 may initially set the Ethernet card as the primary adapter and the PCMCIA adapter as a secondary adapter. While in the middle of downloading a file through the active primary adapter, if the user must go to a meeting in another building, he can simply unplug the Ethernet connection and start on the wireless PCMCIA card. The user is then free to go to the meeting while his laptop continues the downloading session using the wireless LAN connection. The user will be on time at the meeting and be able to finish downloading without any delay.

**[0022]** Further, the NAA 60 can be used in conjunction with Mobile Internet Protocol to allow a mobile device to roam seamlessly between different subnets having different NATs. Similarly, the present invention also applies to networks using various packet based wireless access technologies. As long as there are at least two different NATs, the present invention preserves the integrity of active network applications while providing smooth transition from one NAT to another.

**[0023]** It is noted that in addition to providing switching between different network access technologies, as

mentioned above, the NAA 60 also works in conjunction with other Mobile Internet Protocol functions such as IP-in-IP encapsulation/de-capsulation, proxy ARP, gratuitous ARP, etc.

5 **[0024]** The above disclosure provides many different embodiments, or examples, for implementing different features of the invention. Also, specific examples of components, and processes are described to help clarify the invention. These are, of course, merely examples and are not intended to limit the invention.

10 **[0025]** While the invention has been particularly shown and described with reference to the preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and detail 15 may be made therein without departing from the spirit and scope of the invention.

### Claims

20 1. A method for switching between two different network access technologies on a networked hardware platform without interrupting an active network application, the networked hardware platform sending and receiving information in a data packet form, said method comprising the steps of:

25 determining an active network adapter by monitoring packet traffic and hardware status of one or more network adapters available on the networked hardware platform; and  
30 dynamically engaging the active network adapter by a network access arbitrator to process at least one data packet,  
35 wherein the engagement of the active network adapter is invisible to the active network application.

40 2. The method of claim 1 wherein the network access arbitrator defines a virtual anchor adaptor driver that is known as the only adapter driver to the network application regardless of the existence of other actual adapter drivers available in the networked hardware platform.

45 3. The method of claim 2 further comprising the steps of:

50 assigning a predetermined network adapter as a primary network adapter for providing the network access; and  
55 initially setting the virtual anchor adaptor driver as the network adapter driver associated with the primary network adapter.

4. The method of claim 3 further comprising the steps of:

changing a hardware destination address of an

incoming data packet to a data link layer address of the primary network adapter; and changing a source hardware address of an outgoing data packet to the data link layer address of the active network adapter.

5

5. A method for switching between two different network access technologies on a networked hardware platform without interrupting an active network application, the platform sending outgoing data packets and receiving incoming data packets for the network application through at least two network adapters, the network adapters providing accesses to the network hardware platform for executing the network application, the method comprising the steps of:

15

providing a network access arbitrator having a virtual anchor adapter driver;  
assigning a network adapter as a primary network adapter;  
detecting an active network adapter; and  
configuring, by the network access arbitrator, the data packets generated by the active network application to continue the network application when the access to the network hardware platform is switched from the primary network adapter to the active network adapter driver,  
wherein the network application detects only  
20  
the network arbitrator when accessing the networked hardware platform.

25

6. The method of claim 5 wherein the step of assigning further includes the step of initially configuring the virtual anchor adapter driver as the network adapter driver associated with the primary network adapter.

30

7. The method of claim 5 wherein the step of configuring further includes the steps of:

35

changing a source hardware address of a data packet for the outgoing information to a data link layer address of the active network adapter; and  
modifying a destination hardware address of a data packet for the incoming information to that of the primary network adapter driver.

40

8. The method of claim 5 wherein the step of detecting further includes a step of receiving information from at least one network adapter about connection or disconnection status of the network adapter and its adapter driver.

45

9. The method of claim 8 further comprising the steps of:

50

5. providing a timer to trigger a timed event; and determining whether at least one adapter receives or sends data packets during two consecutive timed events.

5

10. The method of claim 5 wherein the step of detecting further includes the step of detecting whether the primary network adapter is active.

10

11. A method for switching from a first network access technology to a second network access technology on a networked hardware platform without interrupting an active network application using a network access arbitrator, the active network application sending outgoing information and receiving incoming information in data packets through the networked hardware platform, the first network access technology using a first network adapter driver and the second network access technology using a second network adapter driver, the method comprising the steps of:

15

utilizing the first network access technology for executing the active network application; and selecting the second network access technology for continuing the active network application without interrupting the network application through a network access arbitrator by arbitrating between the first network adapter driver and the second network adapter driver for sending the outgoing information and receiving the incoming information.

20

12. The method of claim 11 wherein the network access arbitrator has a virtual anchor adapter driver that is visible to the active network application.

25

13. The method of claim 12 wherein the step of utilizing further includes the steps of:

30

selecting the first network adapter as a primary network adapter; and  
configuring the anchor adapter driver to be associated with the first network adapter.

35

14. The method of claim 11 wherein the step of selecting further includes the steps of:

40

detecting when the second network adapter driver is active; and  
modifying a source hardware address of a data packet for the outgoing information to be a data link layer address of the second network adapter driver; and  
modifying a destination hardware address of a data packet for the incoming information to be a data link layer address of the first network adapter driver.

45

50

55

15. The method of claim 14 wherein the step of detecting further includes a step of receiving information from the second network adapter about connection or disconnection status of the second network adapter.

5

16. The method of claim 15 wherein the step of receiving further includes the steps of:

providing a timer to trigger a timed event; and determining whether the second adapter receives or sends data packets during two consecutive timed events.

10

17. A system for switching between two different network access technologies on a networked hardware platform without interrupting an active network application, the platform sending outgoing data packets and receiving incoming data packets for the network application through at least two network adapters, the network adapters providing access to the network hardware platform for executing the network application, the system comprising:

15

means for assigning a network adapter as a primary network adapter;  
 means for detecting an active network adapter;  
 and  
 a network access arbitrator having a virtual anchor adapter driver for configuring the data packets generated by the active network application to continue the network application when access to the network hardware platform is switched from the primary network adapter to the active network adapter driver;  
 wherein the network application detects only the network arbitrator for accessing the networked hardware platform.

20

18. The system of claim 17 wherein the means for assigning further includes means for configuring the virtual anchor adapter driver initially as the network adapter driver associated with the primary network adapter.

25

19. The system of claim 17 wherein the network access arbitrator further includes:

40

means for changing a source hardware address of a data packet for the outgoing information to a data link layer address of the active network adapter; and  
 means for modifying a destination hardware address of a data packet for the incoming information to that of the primary network adapter driver.

45

20. The system of claim 17 wherein the means for

detecting further includes a means for receiving information from at least one network adapter about connection or disconnection status of the network adapter and its adapter driver.

21. The system of claim 20 further comprising the steps of:

providing a timer to trigger a timed event; and determining whether at least one adapter receives or sends data packets during two consecutive timed events.

22. The system of claim 17 wherein the step of detecting further includes the step of detecting whether the primary network adapter is active.

23. A computer program for switching between two different network access technologies on a networked hardware platform without interrupting an active network application, the networked hardware platform sending and receiving information in a data packet form, said computer program comprising:

instructions for determining an active network adapter by monitoring packet traffic and hardware status of one or more network adapters available on the networked hardware platform; and

instructions for dynamically engaging the active network adapter by a network access arbitrator to process at least one data packet, wherein the engagement of the active network adapter is invisible to the active network application.

24. The program of claim 23 wherein the network access arbitrator defines a virtual anchor adapter driver that is known as the only adapter driver to the network application regardless of the existence of other actual adapter drivers available in the networked hardware platform.

25. The program of claim 24 further comprises:

instructions for assigning a predetermined network adapter as a primary network adapter for providing the network access; and  
 instructions for initially setting the virtual anchor adapter driver as the network adapter driver associated with the primary network adapter.

26. The program of claim 25 further comprises:

instructions for changing a hardware destination address of an incoming data packet to a data link layer address of the primary network

adapter; and  
instructions for changing a source hardware  
address of an outgoing data packet to a data  
link layer address of the active network  
adapter. 5

10

15

20

25

30

35

40

45

50

55

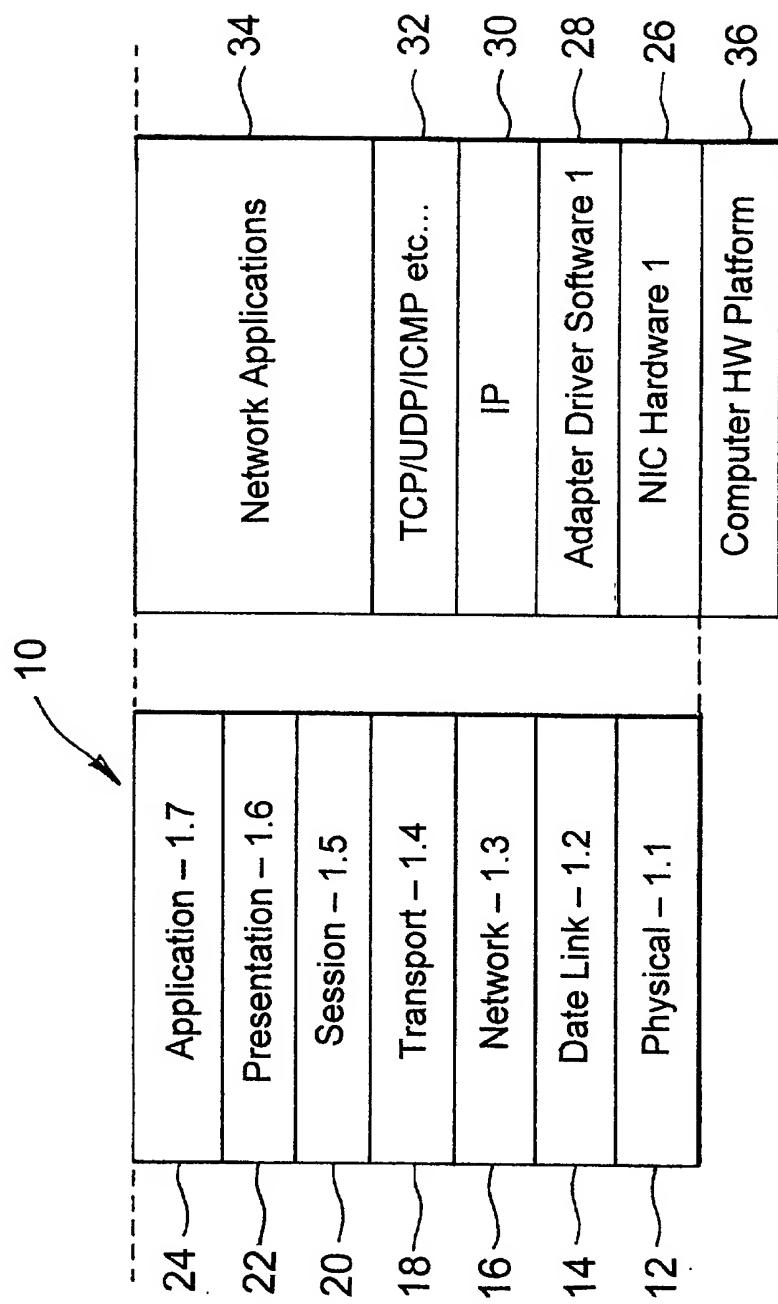


Fig. 1

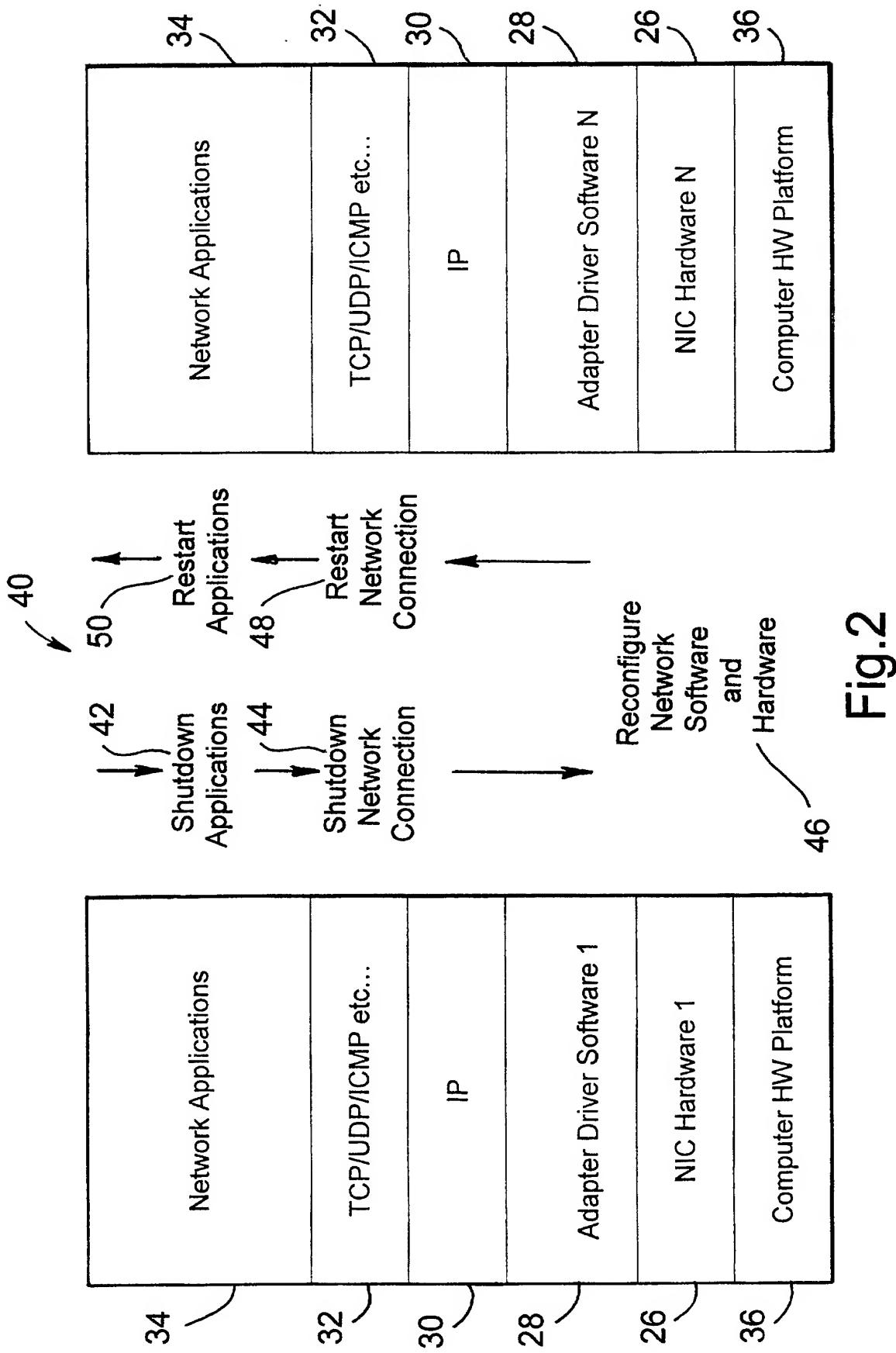


Fig.2

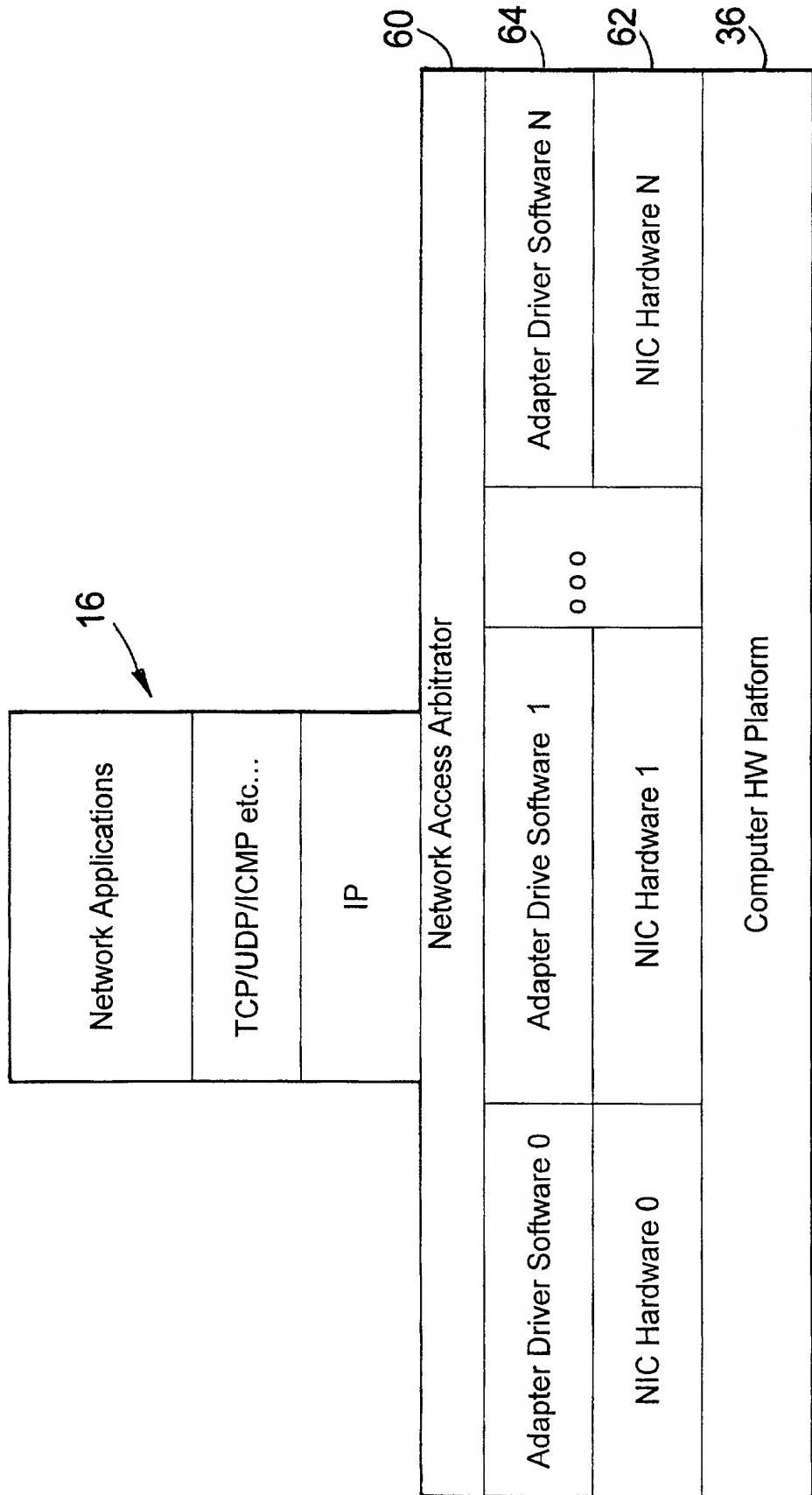


Fig.3